TITLE OF THE INVENTION

SETTING UP A COMMUNICATION PROCEDURE BETWEEN INSTANCES AND A PROTOCOL TESTER USING THE METHOD

BACKGROUND OF THE INVENTION

The present invention relates to protocol testing, and more particularly to a method of setting up a communication procedure between instances, with one of the instances being a protocol tester, and to a protocol tester using the method.

In the field of protocol testing it is necessary to clearly specify a communication procedure by which a test is described so that the procedure may be executed automatically by a protocol tester. Languages such as TTCN (Tree and Tabular Combined Notation) make this possible, but they are complex and difficult to understand for an untrained reader. TTCN has prevailed in the field of Conformance Testing because these tests are very comprehensive, and TTCN supports such comprehensive tests very well. Apart from that, there are various proprietary test description languages. To facilitate understanding a standardized language, MSC (Message Sequence Charts), is used for the purpose of documenting and describing simple procedures. Further details on MSC may be taken from ITU-T Z.120, the contents of which are incorporated herein by reference. MSC consists of standardized process flow diagrams, also referred to as arrow diagrams or X diagrams. These diagrams may be understood independent of programming language. However, automatic execution of communications described by MSC is not possible on protocol testers. To obtain tests that are executable it

is, therefore, necessary to write so-called "scripts", which requires that the user becomes thoroughly acquainted with the relevant programming language. In addition it is necessary to prepare documentation that is generally understood. For a test it is, therefore, necessary on the one hand to prepare graphical and textual documentation and on the other hand a source code or binary code that may be executed.

This state of the art results in a number of disadvantages. It is frequently necessary to convert existing tests, so there is a risk of inconsistencies. The test communication specifications often do not contain information on the configuration, or at least not in a format that may be read by a machine or by man. The different languages often represent proprietary approaches, which differ from equipment to equipment and have to be learned anew. The user is not supported or only receives rudimentary support with protocol knowledge when creating the messages and events.

What is desired is a method, and protocol tester using the method, that overcomes the above-noted disadvantages.

BRIEF SUMMARY OF THE INVENTION

Accordingly the present invention provides a method of setting up a communication procedure between instances, one of which is a protocol tester, by executing the following steps on the protocol tester: selecting instances that are to take part in the communication procedure; selecting a protocol layer on the basis of the communication procedure; selecting abstract communication interfaces of the protocol layer for the communication

procedure; selecting communication data; and automatically setting up through the protocol tester on the basis of the above selections the communication procedure. The selections at any one of the steps may be made graphically with parameters selected being assigned description files that are used in the setting up step.

The objects, advantages and other novel features of the present invention are apparent from the following detailed description when read in conjunction with the appended claims and attached drawing.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWING

Fig. 1 is a plan view of a first graphical user interface (GUI) for a method according to the present invention.

Fig. 2 is a plan view of a second GUI for a method according to the present invention.

Fig. 3 is a plan view of the second GUI of Fig. 2 in another presentation mode.

Fig. 4 is a plan view of the second GUI of Fig. 2 in a further presentation mode.

Fig. 5 is a plan view of a third GUI for a method according to the present invention.

Fig. 6 is a plan view of the third GUI of Fig. 5 in another presentation mode.

Fig. 7 is a plan view of the third GUI of Fig. 5 in a further presentation mode.

DETAILED DESCRIPTION OF THE INVENTION

Fig. 1 shows a graphical user interface (GUI) **10** that allows in a first step graphically selecting instances taking part in a communication procedure. Graphical selection in connection means that a symbol or a text proposal is shown graphically on the GUI, such as on a personal computer (PC) screen, and may be selected by simple activation, i.e., by clicking on it with a "mouse." One of the instances is a protocol tester on which the method as described herein is made available, with the protocol tester in the present case emulating a component, TC_1. Using two buttons, "Add" **12** and "Delete" **14**, a user may add further instances or delete instances listed. In a field **16** the compilation of instances is listed, while in another field **18** the compilation is shown as a diagram. In another field **19** the name of the instance may be selected, and in a further field **20** the instance type is shown. Two buttons, "Back" **22** and "Next" **24**, allow the user to move from one level of the definition of the communication procedure to the next, both in the direction of more detailed specifications and in the direction of higher-level presentations. A "Cancel" button **26** allows leaving a level, meaning that the changes made are reset. A "Help" button **28** offers the user further support.

According to Fig. 2, which shows another representation of the GUI **10**, the present communication procedure has the name Gateway_1, as shown in field **23**. Taking part in the procedure is a first instance TC_1, according to field **25**, and a second instance IUT_1, according to field **27**. According to field **29a** the emulated protocol is of the type "isdn12", with field **29b** offering

further protocols from which to choose. In field **30** various communication procedures that may be chosen for further processing may be offered. Buttons **12, 14, 22, 24, 26, 28** described with reference to Fig. 1 appear again in similar form and with similar functions.

Fig. 3 shows the GUI of Fig. 2 in a different presentation mode, in this case for selecting a Service Access Point (SAP), as shown in field **32a**. In field **32b** there are further SAPs from which to choose. All SAPs shown in field **32b** are offered for the selected emulation "isdn12."

Fig. 4 shows another presentation form of the GUI **10** of Fig. 2, with a format for the communication data (Abstract Service Primitives --ASPs, Protocol Data Units -- PDUs) now being used in a field **34** having so-called Message Pools.

Fig. 5 shows another GUI **36** that provides the user with various types of information in a field **38**. First the instances selected by the user, then the test scenario (Gateway_1) agreed in accordance with Figs. 1-4, and finally the data format (Message Pools).

The following initially only refers to Fig. 6. The GUI **36** shows a large number of buttons **40** that, as is known from word processing or graphics programs, may be clicked by using a mouse. Using these buttons **40** the user may graphically set up the communication procedure in field **42**. Fig. 6 shows the possibility of incorporating codes in the programming language Forth (Draft Proposal ANSI Standard 1994) into a block TE_cfg 44 by using an entry mask **46**. To enter codes in another programming language other entry masks may be used.

Returning to Fig. 5 as an example of a part of the communication procedure shown in field **48**, an instance is awaiting, initially alternatively, the ASPs DL_ESTABLISH_CNF_1 or DL_ESTABLISH_IND_1. Next a timer T_WaitInit with a five (5) second duratioin is started and the elapsing of the time is awaited.

Fig. 7 shows an isdn-PDU "SETUP_1" being incorporated into a flow diagram prepared graphically as a send message. ASPs with PDUs from the Message Pool selected earlier are offered in an entry mask **50**. The PDU selected may be entered into a visually highlighted field **52**. In a field **54** the user is offered further information on the ASP or PDU selected.

In this way it is possible to set up the communication procedure, with preferably all selectable parameters being assigned description files that may be used with each other to automatically set up through the protocol tester the communication procedure to be executed between the instances.

When a code that may be executed is created, there are three interacting components. First the GUI stores the selected parameters, in particular the communication sequence, in an internal structure. Then a compiler translates the selected parameters into temporary files. Finally a linker reads the temporary files and converts them into the selected interpreter script language, such as ANS Forth. During this process the communication procedure as defined by the user is written into a script file.

Annex A1 shows the code automatically generated by the protocol tester for the figures described.